university of groningen

University Medical Center Groningen

University of Groningen

Patterns for Designing Learning Management Systems

Avgeriou, Paris; Retalis, Symeon; Papasalouros, Andreas

Published in:
EPRINTS-BOOK-TITLE

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2003

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Avgeriou, P., Retalis, S., & Papasalouros, A. (2003). Patterns for Designing Learning Management Systems. In *EPRINTS-BOOK-TITLE* University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science.

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license. More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 05-05-2023

Patterns for Designing Learning Management Systems

Paris Avgeriou[1], Symeon Retalis[1], Andreas Papasalouros[2],

1- University of Cyprus
Department of Computer Science
75 Kallipoleos Str., P.O. Box 20537CY-1678 Nicosia, Cyprus
Email: {pavger, retal}@ucy.ac.cy

2- National Technical University of Athens
Department of Electrical and Computer Engineering
Software Engineering Laboratory, 15780 Zografou, Athens, GREECE
E-mail: andpapas@softlab.ntua.gr

Abstract

Learning Management Systems are sophisticated web-based applications that are being engineered today in increasing numbers by numerous institutions and companies that want to get involved in e-learning either for providing services to third parties, or for educating and training their own people. Even though the construction of such systems has been taking place for many years, they are still designed and developed from scratch. The reason is that experience from previous Learning Management Systems, is not codified or documented, resulting in forcing the development teams to 're-invent the wheel'. This paper presents an approach of recording design experience in the form of design patterns for Learning Management Systems and aims at developing a pattern language for these systems. The proposed design patterns are semantically organized and categorized, and form the basic core of a pattern language for Learning Management Systems.

KEYWORDS: Design Patterns, Pattern Language, Patterns System, Learning Management Systems, e-Learning, Learning Technology Systems.

1 Introduction

A **Learning Management System (LMS)** is aimed at managing an e-learning environment, establishing the organization and delivery of content, administrating resources and tracking learning activities and results (Collier, 1996; Oleg & Liber, 1999). LMS that are in use today are either commercial products (e.g. WebCT, Blackboard, TopClass), or free open source products (e.g. ILIAS, Manhattan Virtual Classroom), or customized software systems that serve the instructional purposes of particular organizations. LMS that belong to the third category are exponentially increasing, as most education and training institutions are building or planning to build their own LMS. This is due to the fact that a customized LMS will fit better their specific educational/learning purposes, and proves to give a good return of investment over the years. However, the design and implementation of such systems is not an easy task, since they are complex systems that incorporate a variety of organizational, administrative, instructional and technological components (Moore & Kearsley, 1996; Carlson, 1998). Therefore systematic, disciplined approaches must be devised in order to

leverage the complexity and assortment of LMS and achieve overall product quality within specific time and budget limits. One such approach is the use of design patterns, so that these systems will not be designed and implemented from scratch, but based on reusable design experience gained over several years of try-and-error attempts. The need for design patterns and pattern languages in the domain of Learning Management Systems is gradually being accepted by the LMS community and in general, patterns are emerging for various aspects of e-Learning.

It is well documented that patterns do need a fairly rich sample of applications in order for them to be discovered and considered as widely adopted design solutions. Fortunately, the domain of LMS is over a decade old and we are presently witnessing the maturation of this domain. The plethora of LMS offered today to the market of educational or training organizations and the booming development of Learning Management Systems is an irrefutable proof of this ripening. Thus, researchers and developers have the opportunity to look into a vast number of these systems and 'mine' patterns by reverse-engineering the systems that embed good design in order to make that design explicit, and be able to communicate it to other designers, so that it becomes common practice. As a result, good design experience will be codified and a more systematic development process for these systems will be provided to education and training organizations.

Even though the e-learning community is aware of patterns and their advantages, it has not yet initiated an attempt to establish a formal repository of patterns for its own domain. This paper aims to move research steps towards that direction by proposing an initial set of design patterns for Learning Management Systems. The patterns in this paper are meant to work synergistically and become the first part of a *pattern language* for Learning Management Systems. Of course a full pattern language for LMS, as in any other domain, would theoretically consist of a vast collection of patterns, thus it is stressed that the patterns in this paper comprise only a small subset of the entire language. They are a first attempt to formally describe them and initiate the development of the entire language. As in other pattern languages proposed, the patterns of this paper are not new and certainly not innovative, they have been incorporated in LMS for years now. Instead the intention of this form of pattern languages is merely to capture design expertise and present it in a comprehensible and usable format (Lyardet et al., 1999). In this way, designers of new or existing LMS, especially inexperienced designers, can take advantage of previous design expertise and save precious time and resources.

Relevant research work has been conducted in (Lyardet et al., 1998), where the authors propose hypermedia design patterns that can be applied in educational multimedia applications. There is also a repository of patterns in the conventional learning and pedagogical discipline, mainly focused on teacher-based learning (Pedagogical Patterns Project site, 2002). A first approach for initiating a pattern language for Learning Management Systems is described in (Avgeriou et al., 2003).

The structure of this paper is as follows: Section 2 analyses the context of LMS for the application of design patterns by emphasizing on its unique characteristics that affect the definition of the patterns. Section 3 contains a catalog of patterns and the way that the patterns are organized, dividing them into more generic ones and others that are domain specific. In Section 4 some domain specific patterns are described according to a particular template. Finally Section 5 wraps up with conclusions and ideas for future work.

2 The context of LMS Design Patterns

Learning Management Systems have been widely adopted by institutions and instructional designers in order to fulfill certain needs and requirements in a field of ever increasing demands for effective, fast and pedagogically correct education and training. The users of LMS can be classified into three categories:

- The *learners* that use the system in order to participate through distance (in place and/or time) to the educational process. In fact, the learners are the focal users of LMS, in the sense that these systems are being developed in order to satisfy some of their needs and resolve their problems.
- The *instructors*, being the teachers and their assistants that use the system in order to coach, supervise, assist and evaluate the learners (e.g. notify for important issues on an electronic notice board, engage in discussions in electronic fora, communicate and exchange personal messages with learners, collect, assess and return deliverables, etc.).
- The *administrators* of the system, who undertake the support of all the other users of the system and safeguard its proper operational status.

According to (McCormack & Jones, 1997), an LMS offers services for satisfying specific instructional needs and/or automating (partially or fully) instructional events. LMS should support the development and execution of four basic tasks via a simple, friendly and uniform user-interface:

- *Information distribution*, e.g. announcing the tips of the day, calendar, glossary, etc.
- *Management of learning material*, e.g. customisation of the user interface to the needs of the instructor, updating the learning material, etc.
- *Offer of multiple communication facilities*, e.g. asynchronous and synchronous communication.
- *Class management*, e.g. on-line marking of learners' assessments, tracking learners' participation, management of learners' profiles, etc.

The above basic tasks that LMS need to carry out can be further decomposed into smaller sub-tasks, and the latter can be expressed in the form of functional requirements. Therefore, if we consider the functional requirements as problems that designers of LMS have to solve, we can find the appropriate patterns in existing LMS that illustrate the solution to these problems. The methodology used in this paper for 'pattern mining' is governed by such a philosophy. In particular, we first identified the functional requirements of LMS and then tried to discover these features in a number of real LMS that are broadly used. If these features were indeed found in at least three or four LMS, then these features were considered widely adopted and applicable and were therefore regarded as LMS design patterns. The set of LMS that we used to mine the patterns is shown in Table 1. The next step was to describe them in a suitable format in a similar way to patterns of other domains.

Table 1: LMS that were searched for patterns

Name	**Company**	**URL**
WebCT	WebCT, Inc	http://www.webct.com
CoSE	Staffordshire University	http://www.staffs.ac.uk/COSE
LearningSpace	Lotus	http://www.lotus.com/home.nsf/welcome/learnspace/
BlackBoard	Blackboard	http://www.blackboard.com
TopClass	WBT Systems	http://www.wbtsystems.com
VirtualU	Virtual Learning Enviroments	http://www.vlei.com
FirstClass	Centrinity	http://www.firstclass.com

Zebu	Centrinity	http://www.mc2learning.com
Learnlinc	Mentergy	http://learnlinc.com
Intralearn	Intralearn	http://www.intralearn.com
Saba	Saba Software	http://www.saba.com
FLE	UIAH Media Lab	http://fle3.uiah.fi
Convene	Convene	http://www.convene.com
Gentle WBT	Hyperwave AG	http://wbt-2.iicm.edu

As eloquently stated in (Gamma et al., 1994), a classic book that introduced the notion of patterns in the field of Software Engineering, also known as the GOF book, it is more difficult to describe patterns than to actually find them. Almost all of the approaches that have proposed patterns in a subject field, have also suggested a novel way of describing and cataloging them. Our suggestion for a pattern description format lies between the Alexandrian template and the GOF format and contains the following fields:

i. **Name** – a unique name to distinguish the pattern and uniquely refer to it.
ii. **Problem** – a brief description of the design problem at hand.
iii. **Motivation** – an explanation of the origins of the problem, probably with an example for better communicating it. It may also contain the context of the particular problem if it is necessary in order to make it more comprehensible.
iv. **Solution** – a description of the solution proposed by this pattern that addresses the problem and motivation stated earlier.
v. **User category** – one of the three categories of LMS users defined above.
vi. **Known uses** – examples of the pattern in real LMS. This is an important attribute of a pattern since it is claimed that a proposed pattern gets accepted by the corresponding pattern community, only if there have been two or three examples of its use by someone other than the one who suggested the pattern (Buschmann et al., 1996).
vii. **Related Patterns** – other patterns that are related to this one in some way. It is noted that the patterns proposed in this paper, except for being related to each other, are also related to hypermedia design patterns.

The above format is a rather abstract way of describing patterns, as it does not delve into implementation details, but rather expresses a generic solution. The reason for choosing it, was that the patterns found in Learning Management Systems do not contain many implementation details, but are rather generic and abstract and can be implemented in several different ways. In addition we do not wish to deal with implementation issues because the technologies are changing too fast and if we attempt to propose specific technologies, they will soon become obsolete. The same practice is used in the hypermedia patterns (Rossi et al., 1997; Garrido et al., 1997; Rossi et al., 1999; Garzotto et al., 1999), as well as the HCI patterns (HCI patterns web site, 2002).

3 Organizing the patterns

Patterns in the various domains are usually categorized in different thematic groups that solve similar problems. Such a categorization assists in managing the patterns, especially when their number increases to a large extent. For the domain of LMS, we have defined five such categories:

1. **Access patterns,** that concern the ways that users may access the various resources of Learning Management Systems.
2. **Learning patterns,** that concern the support of learning processes, offered by instructors to learners.

3. **Instructional patterns**, that concern the various tasks that instructors perform in order to create and edit courses and learning resources.
4. **Informational patterns,** that concern the ways that users are informed of various issues that interest them.
5. **Administrational patterns,** that concern the administration of the LMS per se.

Orthogonally to the above categorization, we add another criterion of distinction: whether the patterns are **generic** patterns that can be applied in any hypermedia system, or whether they are **domain specific** patterns for the domain of LMS. For the case of generic patterns, that apply to all hypermedia systems, they will not be analyzed, since they already have been described in corresponding hypermedia systems literature, like (Hypermedia Design Patterns Repository, 2002) or (Duyne et al. 02). For the case of the domain specific patterns, i.e. patterns that attempt to solve problems particular to LMS, and as such, have not been addressed by hypermedia design patterns, they will be formally described according to the template given in the previous section.
The relation between the pattern categories, the patterns themselves, the functional requirements that they implement and whether they are generic or domain specific, is illustrated in Table 1. As aforementioned the category of a pattern is orthogonal to whether the pattern is generic or domain-specific. However, it is obvious that patterns that belong to each category are either mostly generic or domain specific.

Table 1 – Mapping between the functional requirements and the corresponding LMS patterns

Pattern category	LMS Pattern	Functional Requirement	Type
Access	Personalization	Each user must be provided with a personalized user interface, according to the courses he/she is involved	Generic
	Registration-authentication-access control	Users must be able to register and have access rights according to their role	Generic
	Pervasive references	Users must be able to have access to various resources that are semantically related.	Generic
Informational	Course Announcements	Users must be informed about issues of interest to them	Generic
	Online support	Users must be assisted through a helpdesk for the various system's functions	Generic

	Calendar	Users must be informed about the dates that various events are planned to take place	Generic
Learning	Study toolkit	Learners must be assisted in studying the learning resources in a similar way as they study off-line material	Domain specific
	Glossary	Instuctors must be able to provide a glossary of terms found within the learning resources.	Domain specific
	Management of on-line questionnaires	Instructors must be able to assess the learners through appropriate tests	Domain specific
	Student group management	Learners must be organized in working groups and provided with facilities to work in team projects.	Domain specific
	Asynchronous collaborative learning	Users must be able to asynchronously communicate and collaborate	Domain specific
	Synchronous collaborative learning	Users must be able to synchronously communicate and collaborate	Domain specific
	Searching	User must be able to search for and locate learning resources, easily and efficiently	Domain specific
Instructional	E-book delivery	Instructors must be able to synthesize the form, content and functionality of electronic books.	Domain specific
	Course Creation and Customization	User must be able to easily and semi-automatically create and customize courses	Domain specific
	Web page editing	Instructors must be assisted in creating and editing web pages	Generic

	Student tracking	Instructors must be able to track the learner's progress and learners must be able to be informed of their own progess.	Domain specific
	Student Assignments Management	Instructors must be able to give description of assignments to learners and then receive them on-line	Domain specific
Administrational	Course Initialization	Administrators must be assisted in automatically initializing new courses.	Domain specific
	Course backup – restore	Data of each course should be safely kept.	Generic

The organization of these patterns can also be achieved according to how they reference each other in the 'related patterns' field of their description (Gamma et al. 94). In the most fundamental repositories of patterns such as (Buschmann et al. 96, Gamma et al. 94), graphs or maps are designed that show how the distinct patterns refer to each other and what the nature of their relationship is. Figure 1 depicts the relationships between the proposed LMS design patterns, according to the UML notation. Every pattern is designed as a UML class, while the relations between classes are designed as UML relationships. The semantics of such a relationship is that two related patterns can be combined together, so that designers can take both patterns into consideration. For example 'calendar' and 'course announcements', are related because when a new entry is inserted in the calendar, it can also be proclaimed as a course announcement to all those interested. It is noted that the patterns 'Online Support' and 'Pervasive References' are not shown in this diagram since, these two patterns are horizontal, i.e. they are related to all the other patterns. Domain-specific patterns are rendered yellow, while generic patterns are rendered with a green color.

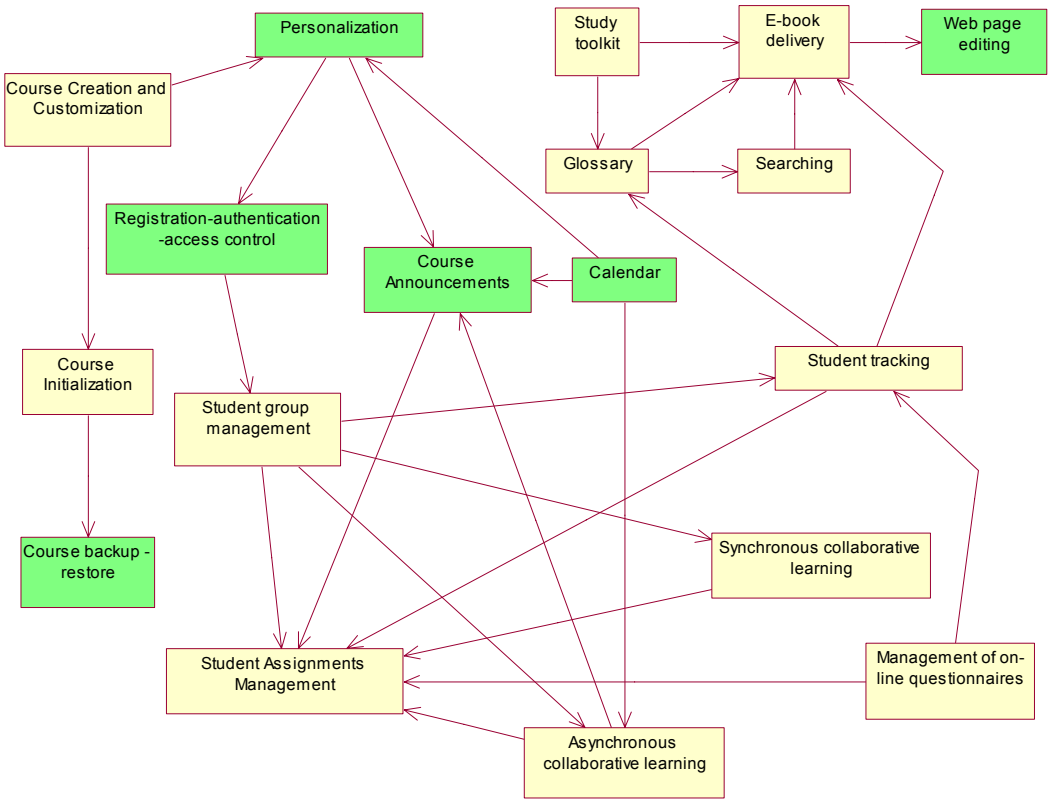

Figure 1 – Design patterns and their relationships

4 Some domain specific patterns

In this section, some of the domain specific patterns will be presented, as a first attempt to formally describe them using the template mentioned in section 2. As aforementioned, the generic patterns have already been described in the literature of hypermedia systems design pattern, such as (Hypermedia Design Patterns Repository, 2002) or in (Duyne et al. 02). For example:

- the "Pervasive references" pattern, that refers to how users can have access to various tools of the LMS from parts not directly related to them, is related to the *Landmark* hypermedia design pattern described in (Rossi et al., 1999), which describes the same mechanism for providing easy access to different though unrelated subsystems in a hypermedia application.
- The "course announcements" pattern is related to the *News* hypermedia design pattern described in (Rossi et al., 1999), which describes the same mechanism for providing the latest news about a particular company in commercial web sites
- The pattern called "searching" that is used for looking for learning resources using learning object metadata, is related to patterns such as 'Selectable Search Space', 'Selectable Keywords', 'Structured Answer', 'Selectable Search Engine' and 'Simple Search Interface' (Lyardet et al., 1999) which are relevant for providing guidelines on how to make effective search engines for Web Information Systems.

4.1 Learning Patterns

4.1.1 Study toolkit

i. **Problem:** how can the learners be assisted in studying the learning resources instead of being limited to reading simple HTML pages?

ii. **Motivation:** There are many facets to this problem. A first one is that most learners find it difficult to study on-line material because they are used to particular methods of studying paper-based courseware and can't get accustomed to reading from the screen passively. When reading paper-based material, learners usually underline or highlight words or phrases, place bookmarks on particular pages, make annotations on the side etc. These functions obviously can't be performed on a plain web page and they need to be incorporated as an explicit service of the LMS. Moreover, instructors often wish to mark or make annotations on students' assignments or deliverables or even web pages of the learning material in order to pinpoint some critical issues and disseminate either publicly or privately to the learners. Another facet of this problem is that learners can't remain connected to the server for many hours for financial reasons (e.g. connection through a dial-up modem) or because they have problems with their connection (limited bandwidth, server down, network congestion). In this case the learners need to download the learning material, store it locally on their computer and use it whenever they want to. Of course this is not a simple download problem, since the learning material may be comprised of numerous pages, linked implicitly through the LMS navigational mechanisms, may have an LMS-made table of contents etc. Finally another facet of this problem is that learners do not want to do on-line studying at all and would rather print the material and read it from paper. Once again this is not a simple download problem, as described earlier.

iii. **Solution:** Provide a study toolkit for the learners to use, which will facilitate them in studying the courseware according to their own preferences. This tool should offer them a set of tools that allow the user:

- to underline, strikethrough and highlight sentences using various color pens for creating annotations on the text
- to put bookmarks on point of interest and/or make comments within the hypertext using either "free text" or specific notations, i.e. a specific symbol should mean "question mark", "criticism", etc.
- to add annotations in any format (text, image, hyperlink, audio, video)
- to characterize an annotation as private or public
- to make queries or short queries according to date, the author, the annotation type.

The annotation tools should also allow the user to 'compile' the learning material in such a format that can be downloaded and stored locally, and which will allow them to add annotations or comments that could be easily "uploaded" to the LMS.

iv. **User category:** Learners and instructors.

v. **Known uses**: WebCT, VirtualU, Blackboard, CoSE, Intralearn, TopClass, LearnLinc, FirstClass and LearningSpace provide the ability to set bookmarks, while CoSE, Intralearn, FirstClass and LearningSpace provide annotation tools but with less functionality as the one described above. WebCT and BlackBoard provide the tools for 'compiling' the learning content in a downloadable and printable format.

vi. **Related Patterns:** E-book delivery, assignments.

4.1.2 Management of on-line questionnaires

i. **Problem:** How can web-based quizzes be created, delivered and graded?

ii. **Motivation:** One of the main learning activities of the instructional process is students' assessment. Assessment is one of the main mechanisms for checking and monitoring students' level of knowledge. It is very beneficial for the instructor to assign particular questions to learning units where the student should check the knowledge she/he is supposed to have obtained. Assessment can be automated in order to save instructors' time and effort in delivering and grading tests. Automation also offers to learners the ability to perform assessment without any time and place constraints. However, the on-line administration (creation, delivery and grading) of tests for the assessment of students is a complicated task. The "Question and Test Interoperability" IMS standard acts as a guide for the creation of assessment tests.

iii. **Solution:** The system should enable the instructors:

- to create on-line both closed-end questions with predefined answers, that are able to be automatically graded and open-end questions, that need to be graded by an instructor
- to create/edit on-line closed-end questions of various types: multiple choice, fill-in the blanks, etc. and easily mention the corresponding right and wrong answers. The hint messages and/or feedback messages that will be shown to the student in case of wrong and/or right answer should be stated
- to administer the delivery of the online test. More specifically, the instructor should be able to state how many times an online test can be answered by the student, the duration of the assessment (time limits), to announce the schedule of on-line tests as well as their grading so that students get informed on time
- to be able to allocate a grade to each question of a test separately and/or to the whole test updating the students' records
- to search for possible questions, that could be integrated into a newly made test, in a pool of already made online tests. In some cases it is valuable to incorporate into a LMS a ready made questionnaire that appears in another LMS. Conformance to QTI standard for question interoperability is necessary in this case.

The system can optionally support adaptive question sequencing, customizing the succession according to which the questions are given to the learner. The answer to a particular question (right or wrong) might change the sequence of the test questions and the related study material according to specific sequencing rules.

iv. **User category:** Instructor, Learner.

v. **Known uses: A**ll LMS that were reviewed have some mechanism for on-line questionnaires.

vi. **Related Patterns:** Assignments management, Student tracking

4.1.3 Student group management

i. **Problem:** How should groups of students be created and managed, and how can projects be assigned to these groups?

ii. **Motivation:** One of the most complicated tasks of both traditional and on-line courses is the management of groups of students. Students must be grouped in working teams,

their progress should be tracked during the project time, and ways of communication between the members of the group and the supervising instructor must be established. In addition there must be some repository for the artifacts of the projects assigned to these groups and a mechanism for grading the students.

iii. **Solution:** Provide a tool for the creation of groups of students. The groups can be created either manually, by the instructors, or automatically by the system. The tool should also provide the ability to assign projects to groups, and, optionally, allocate space for the project deliverables, as well as provide a mechanism for the easy upload of these deliverables from group members. The communication between the members of the group should be established through asynchronous (e-mail, discussion forums) or synchronous (chat, video conference) mechanisms. The system should permit the supervisor of each project to participate in the communication sessions between the members of the groups, to track their progress by reviewing the artifacts of the project and to grade each student at the end of the project.
More specifically the instructor should be able:
- To announce the subjects of the assignments as well as to specify related learning resources (either online or offline) and ask the learners to form groups and choose subject (in case of more than one)
- To see conflicts in the students' choices (e.g. more than one group has chosen the same subject)
- To accept or reject the students' selection of subject. In the latter case, he or she should be able to allocate other subject to them. Moreover, the instructor could be able to manually change the synthesis of the group
- To communicate with the members of the group. The contact information of the group members should be extracted from the LMS database
- To mention whether the students' deliverables will be publicly available or not
- To grade the students' deliverables

The learner should be able:
- To access the proposed subjects of the assignments and get informed about allocations up to that point
- To choose a proposed subject of the assignments and state the rest of the group members
- To upload the deliverables for the assignments.

iv. **User category**: Learner, Instructor.

v. **Known uses:** Blackboard, CoSE, FirstClass, Convene, LearningSpace and WebCT provide tools for the creation and the management of workgroups of students. Gentle WBT has a tool for the definition of working groups, which is available to all types of users.

vi. **Related Patterns:** Student Assignments Management, Asynchronous collaborative learning, Synchronous collaborative learning, Student tracking, student toolkit

4.1.4 Asynchronous collaborative learning

i. **Problem:** How to allow and facilitate learners and instructors to asynchronously collaborate and interact, to engage learners in problem-solving and critical thinking about issues in a domain, to be able both to mentor and to assess these interactions?

ii. **Motivation:** When students work together they learn from one another and extend their interaction and learning outside of class. Busy schedules and commuting students often

make group work difficult to coordinate. When properly applied, technology can eliminate these barriers to collaboration. The main goals for asynchronous collaboration are:

- to provide a comfortable setting for contribution by all group members
- to enable convenient collaboration without restrictions of time or place
- to archive interactions between learners and instructors

iii. **Solution:** Asynchronous computer mediated communications (ACMC) can effectively and efficiently support the asynchronous collaborative learning process, due to the fact that they offer flexibility in the use of time as well as space. The most common type of ACMC is the asynchronous text-based communication, such as e-mail, mailing lists, web-based discussion fora.

iv. **User category:** Learners and instructors

v. **Known uses:** All LMS provide both customized e-mail client-servers and discussion fora. Most of them offer tools for creating group mailing lists.

vi. **Related Patterns:** Personalization, Synchronous collaborative learning, Student group management, Student Assignments Management, Announcements, Information distribution, Synchronous collaborative learning.

4.1.5 Synchronous collaborative learning

i. **Problem:** How to allow and facilitate learners and instructors to interact synchronously, collaborate and co-operate with peers?

ii. **Motivation:** Synchronous collaborative learning is a computer-mediated effort that simulates face-to-face interaction. Since body language and facial expressions cannot be conveyed through asynchronous communication, real-time communication allows contributions participation, sharing information and social dialogue at a distributed environment. The main advantages of synchronous multimedia communication are:

- "Next best thing to being present at a lecture hall"
- Very visual medium; students and teachers can begin to relate to one another.
- Good for distance education novices for developing a "learning community"

iii. **Solution:** Synchronous multimedia communication tools make it possible for learners and instructors at different sites to partake in the same conference at the same time through the "magic" of two-way audio and two-way compressed video. Examples of types of synchronous communication include:

- text-based Internet chats
- instant messaging
- audio & video conferencing
- virtual whiteboard applications
- shared applications

iv. **User category:** Learners and instructors

v. **Known uses:** All LMS provide some sort of chat or conferencing service.

vi. **Related Patterns:** Asynchronous collaborative learning, Student group management, Student Assignments Management

4.2 Instructional Patterns

4.2.1 E-book delivery

i. **Problem:** How can the instructors be facilitated with an easy and consistent way of creating and structuring electronic course books using hypermedia content?

ii. **Motivation:** No matter what learning theory and instructional design strategy is adopted by the Instructors or Institutions, the dissemination of learning content in the form of a set of web pages delivered over the web is common in every web-based system facilitating learning processes. The learning content must be structured, have consistent style and layout and provide a uniform and self explanatory user interface metaphor allowing its users (Students) to easily navigate into the hypertext.

iii. **Solution:** The system must enable the Instructor to:

- structure the learning content into aggregated logical sets of web pages (i.e. chapters) in a hierarchical manner. These web pages can be uploaded to the system or created from scratch. A run-time system will automatically present the structure content to learners providing appropriate controls for navigation (i.e. next/previous page, next chapter, etc).
- Integrate the actual learning content with other tools related to studying. This is done by associating particular learning resources, i.e. web pages or chapters, to specific tools that manage glossary terms, multiple choice questions, links to other resources, search engines, etc.
- save the created study material in a standardized, interchangeable format, such as the IMS Content Packaging format, so as to be able to reuse the structured content in the same, or different LMS.

iv. **User category:** Instructor**.**

v. **Known uses:** WebCT, Blackboard, VirtualU, COSE, Intralearn, TopClass, LearnLinc, FirstClass, and LearningSpace provide instructors with tools for the creation and management of an electronic book.

vi. **Related Patterns:** Glossary, web page editing, Study toolkit, Searching.

4.2.2 Course Creation and Customization

i. **Problem:** How can the instructors be assisted in building on-line courses in LMS so that some of the tasks they need to perform can be automated?

ii. **Motivation:** LMS need to make the job of instructors easier by providing them with easy-to-use tools for creating, and customizing their courses so that they won't have to be experienced in using the LMS, neither will they have to spend too much time and effort in performing those tasks. This way, courses will not be created from scratch, but instead instructors will reuse some design templates and easily perform generic activities and let the LMS take care of the details. For example if an instructor already has a course named 'Software Engineering: Part I' and wants to create another one for the course 'Software Engineering: Part II' that has roughly the same structure and format, she/he should not create it form scratch. Instead she/he should be able to build the new course by using the old one as a template. Also instructors should not have to perform low-level activities to customize their course but the LMS should provide the appropriate tools. For example if the instructor wants to change the background image

of the course's home page she/he should not change the corresponding HTML tag, but instead set it visually through an LMS tool. Finally courses have to be initialized in the beginning of every semester in an automatic way by resetting student accounts, deleti ng the old announcements, reconfiguring the calendar, cleaning the old file folders etc.

iii. **Solution:** Provide the instructors with appropriate tools for creating a course and customizing it according to their preferences. The creation of courses should be based on design templates with pre-set interfaces, content structure and features or based on existing courses. Instructors should also be equipped with tools to reset the courses on every semester and easily manage the appearance, structure and features of their courses, doing as few things manually as possible.

iv. **User category:** Administrators and Instructors.

v. **Known uses:** WebCT, VirtualU, Blackboard, Intralearn, TopClass, LearnLinc, FirstClass, Convene and LearningSpace provide templates for course creation as well as tools for customizing the various courses characteristics.

vi. **Related Patterns:** Personalization, Course Initialization.

4.2.3 Student tracking

i. **Problem:** How can the instructors track the students' progress while they interact with the LMS 's various features? How can the students be informed of what activities they have already performed in a course?

ii. **Motivation:** In the traditional classroom, instructors watch the students' progress, monitor their various activities, evaluate them and coach them so that they learn as effectively as possible. In the virtual world of LMS, instructors do not interact physically with the students and thus cannot observe them and supervise their learning. For example the instructors do not know whether the students have studied the appropriate learning resources, practiced the on-line exercises, collaborated with their colleagues in their projects, or read the announcements for a course. On the other hand, in large and multifaceted courses, the students do not know which parts of the LMS they have already seen, what remaining tasks do they have to perform etc.

iii. **Solution:** Keep records of the students' activities in terms of which parts of the course they have visited and how long they have spent in them, what tools they have used, and maintain files of the conversations that took place in chat rooms, discussion fora etc. Provide the instructor with a tool for observing these records and facilitate him/her in assessing the various activities that students perform, for example by presenting him with statistics about the students' actions. On the students' side, these LMS services can also provide the students with a log of their personal history so that they know where they have already gone and what remains to be seen.
More specifically, the system must enable the Instructor to:

- check the extend by which a particular learner has accessed the learning material in a specific course
- check whether a student has submitted an his assignments on time or not
- check the degree of participation of a student in collaboration activities i.e. discussion for a, synchronous communication sessions, etc.

iv. **User category:** Instructors and learners.

v. **Known uses:** WebCT, Blackboard, Intralearn, Saba, FirstClass, Convene and LearningSpace provide tools for tracking the progress of students. On the other hand

WebCT, VirtualU, Blackboard, Intralearn, Saba, FirstClass and LearningSpace provide tools for informing students of their own study progress.

vi. **Related Patterns:** E-book delivery, Glossary, Management of on-line questionnaires, Student Assignments Management, Student group management.

4.2.4 Student Assignments Management

i. **Problem:** How to create on-line assignments for students?

ii. **Motivation:** Assigning questions and exercises to students is a common practice for instructors. In the context of a web-based LMS certain matters have to be resolved: How to communicate issues concerning the assignments to students, how to grade students, etc.

iii. **Solution:** Provide a tool for instructors to manage assignments. An instructor can define an assignment adding the following entries: The title of the assignment, a description, links to on-line resources, start and due date. Students are notified for the assignment and prepare their documents for submission. The documents can be sent to the instructor via e-mail.

iv. **User category:** Instructor, Learner.

v. **Known uses:** Virtual-U, WebCT, COSE, Intralearn, Saba, Blackboard, FirstClass, Convene and LearningSpace provide tools for assignments management.

vi. **Related Patterns:** Asynchronous collaborative learning, Synchronous collaborative learning, Announcements, Student tracking. This pattern is also related to the Student Group Management Pattern in the sense that they both facilitate a problem-based instructional approach. The main difference between the two is that while in the former, assignments are disseminated to the whole class and require personal work of each individual student, in the latter, groups are created in order to encourage the collaboration of students along with the supervision of an instructor.

5 Conclusions and future work

This paper has attempted to initiate the establishment of a pattern language for Learning Management Systems and to describe few of the domain specific patterns. We believe that such a pattern language can have the following advantages for designers of Learning Management Systems:

- Reduced time and cost of designing and developing LMS.
- Increased software qualities of LMS and especially in the usability sector.
- Increased pedagogical quality of LMS and especially learning effectiveness.

We have already established a repository of design patterns for Learning Management Systems in order to attract more researchers into depositing their own patterns. This initiative was performed within a E.U. project entitled “E-LEN: A Network of e-learning centers” [http://www.tisip.no/E-LEN/], which aims at producing a repository of patterns concerning various aspects of e-learning. This ongoing work, on the long run, will strengthen the pattern language and offer a wealthy pool of patterns, so that inexperienced designers of LMS can base their work on a sound and systematic basis.

The next step after having defined the patterns is to correlate them to LMS components with well specified software interfaces that could collaborate and interoperate. In the sequel, an experimental LMS, that incorporates the patterns proposed in this paper, will be constructed.

The ultimate aim is to illustrate the actual implementation of these patterns by showing the implementation details and offering a detailed object-oriented description of the patterns.

6 Acknowledgment

We would like to thank our shepherd, Michael Pont for his constructive mentoring, as well as our writers' workshop members for their invaluable feedback during EuroPLOP '03. We also acknowledge that the work described in this paper was performed as part of the E-LEN (A network of e-learning centres) project, which is partly funded under the European Union's Socrates Programme.

7 References

1. Alexander, C. , Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I. and Angel, S. (1977). *A Pattern Language*. Oxford University Press, New York.
2. Alur, D., Crupi, J. and Malks, D. (2001). *Core J2EE Patterns: Best Practices and Design Strategies.* Prentice Hall.
3. Avgeriou, P., Retalis, S., Skordalakis, M., "A Software Architecture for a Learning Management System". *Lecture Notes in Computer Science series, Vol. 2563: Advances in Informatics, Springer-Verlag*, 2002.
4. Avgeriou, P., Papasalouros, A., Retalis, S. and Skordalakis, E., Towards a Pattern Language for Learning Management Systems, IEEE Educational Technology and Society journal, Volume 6, Issue 2, pp. 11-24, 2003.
5. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley, 1998.
6. Borchers, J. (2001). *A pattern approach to interaction design.* John Wiley.
7. Brown, W., Malveau, R., McCormick, H., Mowbray, T. (1998). *AntiPatterns: Refactoring Software, Architectures, and Projects in Crisis.* John Wiley & Sons.
8. Buschmann F., Meunier, R., Rohnert, H., Sommertland P. and Stal, M. (1996). *Pattern-Oriented Software Architecture, Volume 1: A System of Patterns.* John Wiley & Sons.
9. Carlson, P. (1998). Advanced Educational Technologies – Promise and Puzzlement, *Journal of Universal Computer Science (JUCS)*, (Special Issue), 4(3).
10. Collier, G., "Elearning application Infrastructure", Sun Microsystems white paper, http://www.sun.com/products-n-solutions/edu/whitepapers/index.html, January 2002.
11. Cooper, J. (2000). *Java Design Patterns.* Addison-Wesley.
12. Duyne, D., Landay, J., Hong, J. (2002). *The Design of Sites: Patterns, Principles and Processes for Crafting a Customer-Centered Web Experience,* Addison-Wesley.
13. Eriksson, H. and Penker, M. (2000). Business Modelling with UML, Business Patterns at work, John Wiley & Sons.
14. Fowler, M. (1996). *Analysis Patterns: Reusable Object Models*, Addison-Wesley.
15. Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley.
16. Garrido, A., Rossi, G. and Schwabe, D. (1997). Pattern Systems for Hypermedia. In Proc. of *PLOP'97* - University of Illinois, Monticello, USA.
17. Garzotto, F., Paolini, P., Bolchini, D. and Valenti, S. (1999). Modeling-by-Patterns of Web Applications", In Proc. *International Workshop on the World Wide Web and Conceptual Modeling*, WWW CM'99, Paris, November 1999.
18. HCI design patterns web site http://www.hcipatterns.org/

19. Hypermedia Design Patterns Repository, http://www.designpattern.lu.unisi.ch/HypermediaHomePage.htm
20. IEEE Learning Technology Standards Committee (LTSC). (2001a). *Draft Standard for Learning Technology Systems Architecture (LTSA)*, Draft 9, http://ltsc.ieee.org/.
21. IEEE Learning Technology Standards Committee (LTSC). (2001b). *Draft Standard for Learning Object Metadata (LOM)*, Draft 6.4, http://ltsc.ieee.org/.
22. Jacobson, I., Booch, G., Rumbaugh, J.: The Unified Software Development Process. Addison-Wesley, 1999.
23. Lyardet, F., Rossi, G. and Schwabe, D. (1998). Using Design Patterns in Educational Multimedia Applications. In Proc. *EDMedia'98*, Freiburg, Germany.
24. Lyardet, F., Rossi, G. and Schwabe, D. (1999). Patterns for Adding Search Capabilities to Web Information Systems. In Proc. of *Europlop'99*, pp. 134-147, Kloster Irsee, Germany, IEEE Press.
25. McCormack, C. & Jones, D. (1997). *Building a Web-based Education System*, Wiley Computer Publishing.
26. Moore, M. G., & Kearsley, G. (1996). *Distance Education: A Systems View*, Wadsworth Publishing Company.
27. Nanard, M., Nanard, J. and Kahn, P. (1998). Pushing reuse in hypermedia design: golden rules, design patterns and constructive templates. In Proc. of *Hypertext'98*, Pittsburgh, Pennsylvania, USA.
28. Oleg, S., Liber, B.: A framework of pedagogical evaluation of Virtual Learning Environments. Available online at [http://www.jtap.ac.uk/reports/htm/jtap-041.html], 1999.
29. Pedagogical Patterns Project site, http://www.pedagogicalpatterns.org/.
30. The Rational Unified Process, 2000, v. 2001.03.00.23, Rational Software Corporation, part of the Rational Solutions for Windows suite.
31. Rossi, G., Garrido, A. and Carvalho, S. (1996). Design Pattern for Object-Oriented Hypermedia Applications". In Vlissides, J., Coplien, J., Kerth, N. (Eds.) *Pattern Languages of Program Design, Vol. 2*, chapter 11, pp. 177-191., Addison-Wesley, 1996.
32. Rossi, G., Schwabe, D., and Garrido, A. (1997). Design Reuse in Hypermedia Application Development. In Proc. of *Hypertext'97*, Southampton, UK.
33. Rossi, G., Lyardet, F. and Schwabe, D. (1999). Patterns for designing navigable spaces. In Harrison, N., Foote. B. and Rohnert, H. (Eds.) *Pattern Languages of Programs 4*, Addison Wesley.
34. Schmidt, D., Stal, M., Rohnert, H., and Buschmann, F. (2000). *Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects*. John Wiley & Sons.